REMARKS/ARGUMENTS

This Amendment is in response to the Final Office Action dated May 19, 2005. Claims

1-27 are pending. Claims 1-27 have been rejected. Claim 6, 9, 12, and 19 have been amended

to correct typographical errors. Claims 1-27 remain pending. For the reasons set forth more

fully below, Applicant respectfully submits that the claims as presented are allowable.

Consequently, reconsideration, allowance, and passage to issue are respectfully requested.

In the event, however, that the Examiner is not persuaded by Applicant's amendments

and arguments, Applicant respectfully requests that the Examiner enter the amendments and

arguments to clarify issues upon appeal.

Claim Objections

The Examiner has stated:

> **2. Claim 9 is objected to because of the following informalities: "A method for performing for an action: should be –A method for performing an action--.**

In response, claim 9 has been corrected in accordance with Examiner's suggestion to

address the above-referenced objection.

Claim Rejections – 35 USC 102

The Examiner has stated:

> **Claims 1-3, 9, 10, 12, 13, 16, 17, 19, 20, 23 and 27 are rejected under 35 U.S.C. 102(a) as being anticipated by applicants admitted prior art, (AAPA), in the background section of the instant application. The paragraph and line numbers of the PGPUB application are used to cite the reference.**
> **As per claim 1, AAPA discloses a computer system for performing an action on a target model, wherein the target model is associated with a notify model, the target model comprising target objects and the notify model comprising notify objects (¶10:1-11:9, "It is known to those skilled in the art that programming code can be written, especially in object-oriented languages, to receive event notifications (i.e., the programming code is notified (to perform an action) when certain changes occur). Depending on the context, both source models and object models can be either a notify model, or a target model.... (for example), a file in the source model is changed and an object model is modified in**

response to the file change. In this first exemplary scenario the source model is the notify model, and the object model is the target model"), the computer system comprising:

-a model map for mapping the notify objects of the notify model to associated target objects in the target model (¶11:5-9, "(for example), a file in the source model is changed and an object model is (automatically) modified in response to the file change. In this first exemplary scenario the source model is the notify model, and the object model is the target model", and the notify model must be mapped to the target model in order for the cited situation to occur),

-an action operator for performing the action on one or more target objects in the target model in response to a modification of a selected notify object (¶11:5-9, "a file in the source model is changed and an object model is modified, (using an action operator), in response to the file change. In this first exemplary scenario the source model is the notify model, and the object model is the target model", and, ¶12:1-3, "a single instance of a notify model may consist of several models. Similarly, an instance of a target model may also consist of several models")

-wherein, the action operator performs the action on one or more identified target objects associated with the modified selected notify object, the one or more identified target objects being determined with reference to the model map (¶11:5-9, "(for example), a file in the source model is changed and an object model is (automatically) modified, (using an action operator), in response to the file change. In this first exemplary scenario the source model is the notify model, and the object model is the target model", and the notify models must be mapped to the target models in order for the cited situation to occur).

As per claim 2, the rejection of claim 1 is incorporated and further, AAPA discloses that the notify model is a model of an object in an object oriented computer language and wherein the target model is source code associated with the object (¶11:7-9, "In this first exemplary scenario the source model is the notify model, and the object model is the target model").

As per claim 3, the rejection of claim 1 is incorporated and further, AAPA discloses means for generating an event notification signal when the selected notify object is modified, wherein the action operator performs the action responsive to receipt of the event notification signal (¶10:1-11:9, "It is known to those skilled in the art that programming code can be written, especially in object-oriented languages, to receive event notifications (i.e., the programming code is notified (to perform an action, using the action operator) when certain changes occur)").

As per claims 9, 10, 12, 13 and 27, this is a method version of the claimed system discussed above, in claims 1-3, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see AAPA, ¶10:1-12:3.

As per claims 16, 17, 19 and 20, this is a computer readable medium version of the claimed system discussed above, in claims 1-3, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see AAPA ¶10:1-12:3.

As per claim 23, this a product version of the claimed system discussed above, in claims 1-3, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see AAPA, ¶10:1-12:3.

Response to Arguments

Applicants arguments have been considered but they are not persuasive.

In the remarks, the applicant has argued substantially that:

AAPA fails to teach, show, or suggest the recited mapping of notify objects of a notify model to associated target objects of a target model, at p. 12:19-13:4.

Examiner's response:

The examiner disagrees with applicant's characterization of the applied art. AAPA does disclose the recited mapping of notify objects of a notify model to associated target objects of a target model. AAPA, ¶11 discloses several practical examples of the mapping of notify objects of a notify model to associated target objects of a target model.

In the remarks, the applicant has argued substantially that:

The model modification of AAPA fails to teach, show, or suggest such object identification and action through the use of mapping, as occurs in the recited invention at p. 13:6-16.
Examiner's response:
The examiner disagrees with applicant's characterization of the applied art. AAPA discloses object identification and action through the use of mapping, as occurs instant application. AAPA, ¶11 discloses several practical examples of object identification and action through the use of mapping.

Applicant respectfully disagrees with the Examiner's rejections. The present invention provides a method and system for incremental actions relating to notify and target models. In a preferred embodiment, a target model includes target objects, and a notify model includes notify objects. The system and method include a model map for mapping each of the notify objects to an associated target object, and an action operator for performing an action on one or more target objects in the target model in response to a modification of a selected notify object. In the system and method, the action operator performs the action on one or more identified target objects associated with the modified selected notify object, the one or more identified target objects being determined with reference to the model map.

By utilizing the present invention, only the objects in the target model affected by the changes in the notify model are identified. Thus, the action (e.g., a validation check) is performed only on the affected parts of the target model. Advantageously, the incremental nature of this action identifies problems sooner, because the action can be run more often, since, given less input, it will not take as much time to complete. When problems are identified sooner, there is less opportunity for other code to become dependent on faulty behavior, and thus less chance that the fix for faulty behavior will break other code. Consequently, this incremental nature of the action performed results in reduced time and thus increases the efficiency of a developer. Applicants admitted prior art (AAPA) does not teach or suggest these features, as discussed below.

AAPA discloses a target model that changes when a notify model is changed. A "notify

model" is used to represent a model(s) which, when changed, results in a signal, known as an "event notification", being issued which indicates that a change has occurred. A "target model" is used to represent a model(s) which, when a change has occurred in the notify model, either changes in response to the change in the notify model or is changed concurrently with the change in the notify model. The notification of a change only operates in one direction from the notify model to the target model. Applications assist developers in generating code which, for many functions, is relatively easy to complete but quite time consuming to code and test. As a result of the laborious nature of producing some code, automatic code generators have been created. These automatic code generators, which often use a visual representation, or model, of the desired output, generate code which represents the graphical model produced. When a change in a notify model is completed, testing of the model (e.g. validating source code) is conducted against the complete target model. Additionally, testing of the target model (e.g. validation of an object model) is commenced by a user command. This process can be extremely time consuming. Additionally, the user may fail to invoke the testing of the target model. As a result, testing of the code is often delayed to the end of the development cycle. Delaying testing often results in defects, which could have been rectified earlier in the development cycle, but are instead identified late in the development cycle. Unfortunately, the problem is exacerbated when additional code, which depends on the defects in the original code, has been developed in the interim.

However, AAPA does not teach or suggest the combination of "a model map for mapping the notify objects of the notify model to associated target objects in the target model," and "an action operator for performing the action on one or more target objects in the target model in response to a modification of a selected notify object," wherein "the action operator performs the action on one or more identified target objects associated with the modified selected notify

object, the one or more identified target objects being determined with reference to the model map," as recited in independent claim 1. The Examiner has argued that AAPA discloses several practical examples of object identification and action through the use of mapping, referring to paragraph 11 of the specification. However, nowhere does paragraph 11 specifically state that there is a mapping of notify objects of a notify model to associated target objects of a target model. Specifically, paragraph 11 states:

> Depending on the context, both source models and object models can be either a notify model, or a target model. To assist in the understanding of the notify and target models, the following examples illustrate three scenarios. In the first exemplary scenario, a file in the source model is changed and an object model is modified in response to the file change. In this first exemplary scenario the source model is the notify model, and the object model is the target model. In a second exemplary scenario a developer's tool changes an object model which causes new source code files to be created. In this second exemplary scenario the object model is the notify model, and the source model is the target model. In a third exemplary scenario source code is used as input for a tool. The tool generates new source code files, and, due to the existence of new code, an object model is updated. In this third exemplary scenario, the source code is both a notify and target model, and the object model is a target model.

The Examiner fails to point out specifically where the description of AAPA mentions a "model map," as recited in the present invention. The Examiner also fails to point out specifically where the description of AAPA mentions that an "action operator" performs an "action on one or more identified target objects," which are "determined with reference to the model map," as recited in the present invention. In one example, AAPA mentions that "a file in the source model is changed and **an object model is modified** in response to the file change" (page 3, lines 6-8). However, the modification of the object model is only **generally** described. There is no specific teaching as to how the object model is modified. In a second example and in a third example, AAPA mentions that new source code files are created when there are changes in an object model. In both of these examples, there is no specific mention that an existing source model is modified, or if one is modified, how it is modified. Clearly, all three examples

do not specifically teach or suggest a model map that is utilized for the modification or how a model map would be utilized. Clearly, all three examples do not specifically teach or suggest that an action is specifically taken on one or more "identified target objects," which are "determined with reference to the model map," as recited in the present invention.

Furthermore, in accordance with the present invention, action (e.g., a validation check) is performed only on the affected parts of the target model. AAPA teaches that when a change in a notify model is completed, the testing of the model is conducted "against the **complete** target model." Clearly, action on the complete target model as taught by AAPA is different from action taken on only the one or more "identified target objects," which are "determined with reference to the model map," as recited in the present invention.

Accordingly, since AAPA does not teach or suggest the cooperation of elements as recited in the present invention. Therefore, claim 1 is allowable over AAPA.

Independent claims 9, 16, 23, and 27

Similar to independent claim 1, independent claims 9, 16, 23, and 27 recite the "mapping the notify objects of the notify model to associated target objects in the target model," and performing an "action on one or more identified target objects," which are "determined with reference to the mapping. As described above, with respect to independent claim 1, AAPA does not teach or suggest these features. Accordingly, the above-articulated arguments related to independent claim 1 apply with equal force to claims 9, 16, 23, and 27. Therefore, claims 9, 16, 23, and 27 are allowable over AAPA for at least the same reasons as claim 1.

Independent claim 25

Similar to independent claim 1, independent claim 25 recites "performing code validation on one or more identified target objects associated with a selected notify object, the one or more identified target objects being determined with reference to a map of the associations between the notify objects in the notify model and the target objects in the object model." As described above, with respect to independent claim 1, AAPA does not teach or suggest these features. Instead, AAPA teaches that when a change in a notify model is completed, the testing of the model is conducted "against the **complete** target model." Accordingly, the above-articulated arguments related to independent claim 1 apply with equal force to claim 25. Therefore, claim 25 is allowable over AAPA for at least the same reasons as claim 1.

Dependent claims 2-3, 10, 12-13, 17, and 19-20

Dependent claims 2-3, 10, 12-13, 17, and 19-20 depend from independent claims 1, 9, 16, 23, and 27, respectively. Accordingly, the above-articulated arguments related to independent claims 1, 9, 16, 23, and 27 apply with equal force to claims 2-3, 10, 12-13, 17, and 19-20, which are thus allowable over the cited reference for at least the same reasons as claims 1, 9, 16, 23, and 27.

Claim Rejections – 35 USC 103

The Examiner has stated:

> **Claims 4-8, 11, 14, 15, 18, 21, 22, 24-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over applicants admitted prior art, (AAPA), in the background section of the**

instant application in view of Atkinson et al., (Atkinson), U.S. Patent no. 5,613,124.

As per claim 4, the rejection of claim 1 is incorporated and further, AAPA doesn't explicitly disclose that the model map is one of a lookup table and a database.

However, Atkinson, in an analogous environment, discloses that the model map is one of a lookup table and a database (col. 3:36-39, "The virtual function table (i.e. lookup table) contains an entry (which maps a relationship) for each virtual function member defined for the object. Each entry contains a reference to the code that implements the corresponding function member").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Atkinson into the system of AAPA to have the model map as a lookup table or a database. The modification would have been obvious because one of ordinary skill in the art would want to use the well known lookup table or database in order to store and retrieve data, involved in complex relationships, in an organized and efficient fashion.

As per claim 5, the rejection of claim 4 is incorporated and further, AAPA doesn't explicitly disclose that the model map maps portions of the notify objects to associated portions of the target objects.

However, Atkinson, in an analogous environment, discloses that the model map maps portions of the notify objects to associated portions of the target objects (col. 33:31-33, "information that indicates which portion of the object is to be used for generating the (portion) of the presentation data", additionally, the AAPA and Atkinson references are directed toward object oriented technologies, wherein separating objects into their constituent parts, maintaining complex relationships involving objects and their constituent parts, and modification and the propagation of modifications involving objects and their constituent parts (without affecting the rest of the application) are common, well know techniques).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Atkinson into the system of AAPA to have a model map that maps portions of the notify objects to associated portions of the target objects. The modification would have been obvious because one of ordinary skill in the art would want to exploit the numerous advantages of object oriented technologies (e.g. minimizing the time or coding effort required to produce an application relying on the notify and target object relationship).

As per claim 6, the rejection of claim 5 is incorporated and further, AAPA discloses that the action performed by the action operator is performed on the identified portions of the target objects in the target model, the identified portions of the target being determined with reference to the model map (¶11:5-9, "(for example), a file in the source model is changed and an object model is modified, (using an action operator), in response to the file change. In this first exemplary scenario the source model is the notify model, and the object model is the target model", and the portion of the notify model must be mapped to the portion of the target model in order for the cited situation to occur, additionally, the AAPA reference is directed toward object oriented technologies, wherein separating objects into their constituent parts, maintaining complex relationships involving objects and their constituent parts, and modification and the propagation of modifications involving objects and their constituent parts (without affecting the rest of the application) are common, well know techniques).

As per claim 7, the rejection of claim 6 is incorporated and further, AAPA discloses that the notify model is a model of an object in an object oriented language and wherein the target object is source code (¶11:11-12, "In this second exemplary scenario the object model is the notify model, and the source model is the target model").

As per claim 8, the rejection of claim 7 is incorporated and further, AAPA discloses that the action performed is a source code validation(¶18:1-3, "Presently, when a change in a notify model is completed, testing of the model (e.g., validating source code or the EJB in the examples described above, respectively) is (the action that is) conducted").

As per claims 11, 14, 15, 18, 21, 22 and 24, the AAPA/Atkinson combination also discloses such claimed limitations as addressed in claims 4 and 8 above.

As per claims 25-26, this is a product version of the claimed system discussed above, in claim 8, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see AAPA, ¶18:1-3.

Dependent claims 4-8, 11, 14-15, 18, 21-22, and 24-26 depend from independent claims 1, 9, 16, 23, 25, and 27, respectively. Accordingly, the above-articulated arguments related to independent claims 1, 9, 16, 23, 25, and 27 apply with equal force to claims 4-8, 11, 14-15, 18, 21-22, and 24-26, which are thus allowable over the cited reference for at least the same reasons as claims 1, 9, 16, 23, 25, and 27.

Furthermore, with regard to dependent claims 4, 14, and 26, the Examiner has relied upon Atkinson to cure the defects of AAPA, which does not describe that the "model map is one of a lookup table and a database," as recited in the present invention. However, Atkinson merely describes a "virtual function table" (column 3, lines 36-39). While the virtual function table of Atkinson may be interpreted as a look up table, there is no mention as to whether the virtual function table (or even a database) can be utilized as the model map of the present invention, which specifically maps notify objects to associated target objects. As argued above, AAPA does not teach or suggest the model map or how it functions.

Accordingly, AAPA in view of Atkinson does not teach or suggest the cooperation of elements as recited in dependent claims 4, 14, and 26. Therefore, claims 4, 14, and 26 are allowable over AAPA in view of Atkinson for at least these reasons.
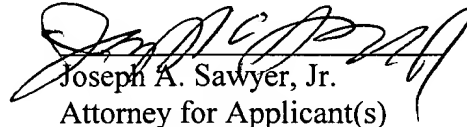
Conclusion

In view of the foregoing, it is submitted that the claims 1-27 are allowable over the cited references and are in condition for allowance. Applicant respectfully requests reconsideration of the rejections and objections to the claims, as now presented.

Applicant's attorney believes this application in condition for allowance. Should any

unresolved issues remain, Examiner is invited to call Applicant's attorney at the telephone

number indicated below.

Respectfully submitted,

SAWYER LAW GROUP LLP

July 6, 2005
Date

Joseph A. Sawyer, Jr.
Attorney for Applicant(s)
Reg. No. 30,801
(650) 493-4540